

A large, stylized letter 'A' is formed using the characters 'S' and 'Y'. The 'S' characters are arranged in a grid-like pattern to form the left and right sides of the letter, while 'Y' characters form the central vertical stem and the horizontal crossbar. The overall shape is a bold, blocky 'A' that fills most of the page.

```

      AAAAAA      CCCCCCCC      CCCCCCCC      000000      UU      UU      NN      NN      TTTTTTTTTT
      AAAAAA      CCCCCCCC      CCCCCCCC      000000      UU      UU      NN      NN      TTTTTTTTTT
AA      AA      CC      CC      CC      00      00      UU      UU      NN      NN      TT
AA      AA      CC      CC      CC      00      00      UU      UU      NN      NN      TT
AA      AA      CC      CC      CC      00      00      UU      UU      NN      NN      TT
AA      AA      CC      CC      CC      00      00      UU      UU      NN      NN      TT
AA      AA      CC      CC      CC      00      00      UU      UU      NN      NN      TT
AAAAA      CC      CC      CC      CC      00      00      UU      UU      NN      NN      TT
AAAAA      CC      CC      CC      CC      00      00      UU      UU      NN      NN      TT
AA      AA      CC      CC      CC      00      00      UU      UU      NN      NN      TT
AA      AA      CC      CC      CC      00      00      UU      UU      NN      NN      TT
AA      AA      CC      CC      CC      00      00      UU      UU      NN      NN      TT
AA      AA      CC      CC      CC      00      00      UU      UU      NN      NN      TT
      CCCCCCCC      CCCCCCCC      000000      UUUUUUUUUU      NN      NN      TT      .:.
      CCCCCCCC      CCCCCCCC      000000      UUUUUUUUUU      NN      NN      TT      .:.
      CCCCCCCC      CCCCCCCC      000000      UUUUUUUUUU      NN      NN      TT      .:.
      CCCCCCCC      CCCCCCCC      000000      UUUUUUUUUU      NN      NN      TT      .:.

LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS

```


ACCOUNT
Table of contents

-- ACCOUNTING MESSAGE SUBROUTINES^{F 13}

15-SEP-1984 23:49:17 VAX/VMS Macro V04-00

Page 0

(1)	36	HISTORY	; DETAILED
(2)	62	DECLARATIONS	
(2)	104	IMAGE ACCOUNTING MESSAGE	
(2)	196	PROCESS ACCOUNTING MESSAGE	
(2)	279	COMMON IDENTIFICATION DATA	

```
0000 1 .TITLE ACCOUNT -- ACCOUNTING MESSAGE SUBROUTINES
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24
0000 25 *
0000 26 *****
0000 27
0000 28 ++
0000 29 FACILITY: EXECUTIVE
0000 30
0000 31 ABSTRACT:
0000 32
0000 33 ENVIRONMENT: KERNEL MODE, IPL 0 OR SPECIAL KERNEL AST (IPL 2)
0000 34
0000 35
0000 36 .SBTTL HISTORY ; DETAILED
0000 37
0000 38 AUTHOR: Steve Forgey CREATION DATE: 01-Aug-1981
0000 39
0000 40 MODIFIED BY:
0000 41
0000 42 V03-005 MLJ0113 Martin L. Jack, 2-May-1983 0:07
0000 43 Always write process termination message to job controller,
0000 44 without filtering on ACMSV_PROCESS. Also fix a broken BSBW.
0000 45
0000 46 V03-004 CWH1002 CW Hobbs 24-Feb-1983
0000 47 Change so that extended pid and owner are used in the
0000 48 accounting records.
0000 49
0000 50 V03-003 LJK0190 Lawrence J. Kenah 15-Nov-1982
0000 51 Fix bug introduced by LJK0186 that did not fully note that
0000 52 registers containing image name and address changed.
0000 53
0000 54 V03-002 LJK0186 Lawrence J. Kenah 22-Oct-1982
0000 55 Perform access checks before picking up image name from
0000 56 image header buffer page in P1 space.
0000 57
```


ACCOUNT
V04-000

-- ACCOUNTING MESSAGE SUBROUTINES H 13
HISTORY ; DETAILED

15-SEP-1984 23:49:17 VAX/VMS Macro V04-00
5-SEP-1984 03:39:51 [SYS.SRC]ACCOUNT.MAR;1

Page 2
(1)

0000 58 :
0000 59 :
0000 60 :--

V03-001 KDM0002
Added \$PRDEF.

Kathleen D. Morse

28-Jun-1982

```
0000 62 .SBTTL DECLARATIONS
0000 63
00000000 64 .PSECT YF$LOWUSE
0000 65
0000 66 :
0000 67 : EQUATED SYMBOLS:
0000 68 :
0000 69 :
0000 70 $ACMDEF ; DEFINE AST CONTROL BLOCK
0000 71 $IPLDEF ; PROCESSOR INTERRUPT LEVELS
0000 72 $MSGDEF ; DEFINE MAILBOX MESSAGE TYPES
0000 73 $PCBDEF ; DEFINE PCB OFFSETS
0000 74 $PHDDEF ; DEFINE PROCESS HEADER OFFSETS
0000 75
0000 76 ASSUME ACMSW_MAILBOX EQ ACMSW_TYPE+2
0000 77 ASSUME ACMSQ_PRVMSK EQ ACMSW_MAILBOX+2
0000 78 ASSUME ACMSL_UIC EQ ACMSQ_PRVMSK+8
0000 79 ASSUME ACMST_USERNAME EQ ACMSL_UIC+4
0000 80 ASSUME ACMST_ACCOUNT EQ ACMST_USERNAME+12
0000 81 ASSUME ACMSB_PROCPRI EQ ACMST_ACCOUNT+8
0000 82 ASSUME ACMSL_PID EQ ACMSB_PROCPRI+4
0000 83 ASSUME ACMSL_STS EQ ACMSL_PID+4
0000 84 ASSUME ACMSL_OWNER EQ ACMSL_STS+4
0000 85 ASSUME ACMST_TERMINAL EQ ACMSL_OWNER+4
0000 86 ASSUME ACMSQ_SYSTIME EQ ACMST_TERMINAL+8
0000 87 ASSUME ACMSQ_LOGIN EQ ACMSQ_SYSTIME+8
0000 88 ASSUME ACMSL_FINALSTS EQ ACMSQ_LOGIN+8
0000 89 ASSUME ACMSL_IMGCNT EQ ACMSL_FINALSTS+4
0000 90 ASSUME ACMSL_CPUTIME EQ ACMSL_IMGCNT+4
0000 91 ASSUME ACMSL_PAGEFLTS EQ ACMSL_CPUTIME+4
0000 92 ASSUME ACMSL_PGFLTIO EQ ACMSL_PAGEFLTS+4
0000 93 ASSUME ACMSL_WSPEAK EQ ACMSL_PGFLTIO+4
0000 94 ASSUME ACMSL_PGFLPEAK EQ ACMSL_WSPEAK+4
0000 95 ASSUME ACMSL_DIOCNT EQ ACMSL_PGFLPEAK+4
0000 96 ASSUME ACMSL_BIOCNT EQ ACMSL_DIOCNT+4
0000 97 ASSUME ACMSL_VOLUMES EQ ACMSL_BIOCNT+4
0000 98 ASSUME ACMSW_NODEADDR EQ ACMSL_VOLUMES+4
0000 99 ASSUME ACMSW_NODENAME EQ ACMSW_NODEADDR+2
0000 100 ASSUME ACMSW_REMOTEID EQ ACMSW_NODENAME+2
0000 101 ASSUME ACMSW_IMAGENAME EQ ACMSW_REMOTEID+2
0000 102 ASSUME ACMSK_PROCLN EQ ACMSW_IMAGENAME+2
```



```
0000 104 .SBTTL IMAGE ACCOUNTING MESSAGE
0000 105
0000 106 :++
0000 107 : FUNCTIONAL DESCRIPTION:
0000 108 :
0000 109 :
0000 110 : INPUT PARAMETERS:
0000 111 : R4 = PCB ADDRESS
0000 112 : R5 = ACB ADDRESS (IF SPECIAL KERNEL AST ROUTINE)
0000 113 : = 0 (OTHERWISE)
0000 114 :
0000 115 : OUTPUT PARAMETERS:
0000 116 : R0 = RETURN STATUS
0000 117 :
0000 118 :--
0000 119
0000 120 .ENABLE LSB
0000 121
0000 122 EXE$IMGPURMSG:: : SEND IMAGE PURGE MESSAGE
OFFE 8F BB 0000 123 PUSHR #^M<R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>;SAVE R1-R11
56 0D 3C 0004 124 MOVZWL #MSG$_PURIMAG,R6 : SET UP MESSAGE TYPE
07 11 0007 125 BRB 10$
0009 126
0009 127 EXE$IMGDELMSG:: : SEND IMAGE TERMINATION MESSAGE
OFFE 8F BB 0009 128 PUSHR #^M<R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>;SAVE R1-R11
56 0C 3C 000D 129 MOVZWL #MSG$_DELIMAG,R6 : SET UP MESSAGE TYPE
0010 130 10$:
16 00000000'EF 01 E1 0010 131 BBC #ACMSV_IMAGE,EXE$GL_ACMFLAGS,20$; IF BC, IMAGE ACCOUNTING NOT ENABLE
11 24 A4 0F E0 0018 132 BBS #PCBSV_NOACNT,PCBSL_STS(R4),20$; IF BS, NO ACCOUNTING FOR PROCESS
00000004'9F 00000000'9F C9 0020 133 SETIPL #IPL$ ASTDEL : LOCK OUT AST DELIVERY
03 12 002C 134 BISL3 @#CTL$GQ_ISTART,@#CTL$GQ_ISTART+4,R1; IMAGE ACCOUNTING ACTIVE ?
002E 135 BNEQ 30$ : IF NEQ, YES
0208 31 002E 136 20$:
57 5E D0 0031 137 BRW EXIT
0031 138 30$:
0031 139 MOVL SP,R7 : SAVE SP (END OF BUFFER)
0034 140 :
0034 141 : IMAGE NAME
0034 142 :
5B 54 D0 0034 143 MOVL R4,R11 : SAVE PCB ADDRESS IN SAFE PLACE
5A D4 0037 144 CLRL R10 : ASSUME NO IMAGE NAME
54 00000000'9F 9A DE 0039 145 MOVAL @#CTL$GL_IMGHDRBF,R4 : GET POINTER TO IMAGE HEADER BUFFER
00000000'GF 16 0040 146 JSB G^EXE$CHRIMAGNAME : IS IMAGE NAME ACCESSIBLE?
0E 13 0046 147 BEQL 35$ : BRANCH IF INACCESSIBLE
5E 53 C2 0048 148 SUBL2 R3,SP : ALLOCATE BUFFER SPACE
7E 53 90 004B 149 MOVB R3,-(SP) : STORE IMAGE NAME SIZE
01 AE 64 53 28 004E 150 MOVC3 R3,(R4),1(SP) : COPY IMAGE NAME
5A 5E D0 0053 151 MOVL SP,R10 : IMAGE NAME ADDRESS
0056 152 :
0056 153 : REMOTE NODE INFORMATION
0056 154 :
59 00000000'9F 9A 0056 155 35$: MOVZBL @#CTL$T_REMOTEID,R9 : GET REMOTE ID BYTE COUNT
12 13 005D 156 BEQL 40$ : IF EQL, NO REMOTE ID
5E 59 C2 005F 157 SUBL2 R9,SP : ALLOCATE BUFFER SPACE
7E 59 90 0062 158 MOVB R9,-(SP) : BYTE COUNT
01 AE 00000001'9F 59 28 0065 159 MOVC3 R9,@#CTL$T_REMOTEID+1,1(SP); COPY REMOTE ID
59 5E D0 006E 160 MOVL SP,R9 : REMOTE ID ADDRESS
```

```

      58 00000000'9F 9A 0071 161 40$:
              12 13 0071 162
              5E 58 C2 0078 163
              7E 58 90 007A 164
01 AE 00000001'9F 58 28 007D 165
      58 5E D0 0080 166
              0089 167
              008C 168 50$:
      51 00000000'9F 9A 008C 169
              12 13 0093 170
              5E 51 C2 0095 171
              7E 51 90 0098 172
01 AE 00000001'9F 51 28 009B 173
      51 5E D0 00A4 174
              00A7 175 60$:
              7E 7C 00A7 176
      54 5B D0 00A9 177
              00AC 178
              00AC 179 : RESOURCE DATA
              00AC 180 :
7E 00000000'9F 00000000'9F C3 00AC 181
      55 00000000'9F D0 00B8 182
      7E 58 A5 00000000'9F C3 00BF 183
      7E 54 A5 00000000'9F C3 00C8 184
      7E 00000000'9F 7D 00D1 185
7E 0108 C5 00000000'9F C3 00D8 186
      7E 4C A5 00000000'9F C3 00E2 187
      7E 38 A5 00000000'9F C3 00EB 188
              00F4 C5 DD 00F4 189
              00000000'9F DD 00F8 190
      7E 00000000'9F 7D 00FE 191
              00B2 31 0105 192
              0108 193
              0108 194
      .DISABLE LSB

      MOVZBL @#CTLST_NODENAME,R8 ; GET NODE NAME BYTE COUNT
      BEQL 50$ ; IF EQL, NO REMOTE NODE NAME
      SUBL2 R8,SP ; ALLOCATE BUFFER SPACE
      MOVB R8,-(SP) ; BYTE COUNT
      MOVC3 R8,@#CTLST_NODENAME+1,1(SP); COPY REMOTE NODE NAME
      MOVL SP,R8 ; REMOTE NODE NAME ADDRESS

      MOVZBL @#CTLST_NODEADDR,R1 ; GET NODE ADDRESS BYTE COUNT
      BEQL 60$ ; IF EQL, NO REMOTE NODE ADDRESS
      SUBL2 R1,SP ; ALLOCATE BUFFER SPACE
      MOVB R1,-(SP) ; BYTE COUNT
      MOVC3 R1,@#CTLST_NODEADDR+1,1(SP); COPY REMOTE NODE ADDRESS
      MOVL SP,R1 ; REMOTE NODE ADDRESS ADDRESS

      CLRQ -(SP) ; NODE ADDRESS, NAME, REM. ID, IMAGE
      MOVL R11,R4 ; RESTORE PCB ADDRESS TO USUAL REGISTER

      SUBL3 @#CTL$GL_IVOLUMES,@#CTL$GL VOLUMES,-(SP); IMAGE VOLUME MOUNTS
      MOVL @#CTL$GL_PHD,R5 ; PHD ADDRESS
      SUBL3 @#CTL$GL_IBIOCNT,PHD$B_BIOCNT(R5),-(SP); IMAGE BUFFERED I/O COUNT
      SUBL3 @#CTL$GL_IDIOCNT,PHD$B_DIOCNT(R5),-(SP); IMAGE DIRECT I/O COUNT
      MOVQ @#CTL$GL_IWSPEAK,-(SP) ; IMAGE WORKING SET AND PAGE FILE PEAK
      SUBL3 @#CTL$GL_IFAULTIO,PHD$B_PGFLTIO(R5),-(SP); IMAGE PAGE FAULT I/O COUNT
      SUBL3 @#CTL$GL_IFAULTS,PHD$B_PAGEFLTS(R5),-(SP); IMAGE PAGE FAULT COUNT
      SUBL3 @#CTL$GL_ICPUTIM,PHD$B_CPUTIM(R5),-(SP); IMAGE CPU TIME
      PUSHL PHD$B_IMGENT(R5) ; IMAGE SEQUENCE NUMBER
      PUSHL @#CTL$GL_FINALSTS ; IMAGE FINAL STATUS
      MOVQ @#CTL$GL_ISTART,-(SP) ; IMAGE START TIME
      BRW IDENT ; JOIN COMMON CODE
```



```
0108 196 .SBTTL PROCESS ACCOUNTING MESSAGE
0108 197
0108 198 :++
0108 199 : FUNCTIONAL DESCRIPTION:
0108 200 :
0108 201 :
0108 202 : INPUT PARAMETERS:
0108 203 : R4 = PCB ADDRESS
0108 204 : R5 = ACB ADDRESS (IF SPECIAL KERNEL AST ROUTINE)
0108 205 : = 0 (OTHERWISE)
0108 206 :
0108 207 : OUTPUT PARAMETERS:
0108 208 : R0 = RETURN STATUS
0108 209 :
0108 210 :--
0108 211
0108 212 .ENABLE LSB
0108 213
0108 214 EX$PRCPURMSG:: : SEND PROCESS PURGE MESSAGE
OFFE 8F BB 0108 215 PUSHR #M<R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>;SAVE R1-R11
56 0B 3C 010C 216 MOVZWL #MSG$_PURPROC,R6 : SET UP MESSAGE TYPE
07 11 010F 217 BRB 10$
0111 218
0111 219 EX$PRCDELMSG:: : SEND PROCESS TERMINATION MESSAGE
OFFE 8F BB 0111 220 PUSHR #M<R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>;SAVE R1-R11
56 03 3C 0115 221 MOVZWL #MSG$_DELPROC,R6 : SET UP MESSAGE TYPE
0118 222 10$:
03 24 A4 0F E1 0118 223 BBC #PCB$_NOACNT,PCB$_L_STS(R4),30$; IF BS, NO ACCOUNTING FOR PROCESS
011D 224 20$:
0119 31 011D 225 BRW EXIT
0120 226 30$:
57 5E D0 0120 227 MOVL SP,R7 : SAVE SP (END OF BUFFER)
0123 228 :
0123 229 : IMAGE NAME
0123 230 :
0123 231 CLRL R10 : NO IMAGE NAME
0125 232 :
0125 233 : REMOTE NODE INFORMATION
0125 234 :
59 00000000'9F 9A 0125 235 MOVZBL @#CTL$_REMOTEID,R9 : GET REMOTE ID BYTE COUNT
17 13 012C 236 BEQL 40$ : IF EQL, NO REMOTE ID
5E 59 C2 012E 237 SUBL2 R9,SP : ALLOCATE BUFFER SPACE
7E 59 90 0131 238 MOVB R9,-(SP) : BYTE COUNT
54 DD 0134 239 PUSHL R4 : SAVE PCB ADDRESS
05 AE 00000001'9F 59 28 0136 240 MOVC3 R9,@#CTL$_REMOTEID+1,5(SP); COPY REMOTE ID
54 8ED0 013F 241 POPL R4 : RESTORE PCB ADDRESS
59 5E D0 0142 242 MOVL SP,R9 : REMOTE ID ADDRESS
0145 243 40$:
58 00000000'9F 9A 0145 244 MOVZBL @#CTL$_NODENAME,R8 : GET NODE NAME BYTE COUNT
17 13 014C 245 BEQL 50$ : IF EQL, NO REMOTE NODE NAME
5E 58 C2 014E 246 SUBL2 R8,SP : ALLOCATE BUFFER SPACE
7E 58 90 0151 247 MOVB R8,-(SP) : BYTE COUNT
54 DD 0154 248 PUSHL R4 : SAVE PCB ADDRESS
05 AE 00000001'9F 58 28 0156 249 MOVC3 R8,@#CTL$_NODENAME+1,5(SP); COPY REMOTE NODE NAME
54 8ED0 015F 250 POPL R4 : RESTORE PCB ADDRESS
58 5E D0 0162 251 MOVL SP,R8 : REMOTE NODE NAME ADDRESS
0165 252 50$:
```

```
51 00000000'9F 9A 0165 253 MOVZBL @#CTL$T_NODEADDR,R1 ; GET NODE ADDRESS BYTE COUNT
      17 13 016C 254 BEQL 60$ ; IF EQL, NO REMOTE NODE ADDRESS
      5E 51 C2 016E 255 SUBL2 R1,SP ; ALLOCATE BUFFER SPACE
      7E 51 90 0171 256 MOVB R1,-(SP) ; BYTE COUNT
      54 DD 0174 257 PUSHL R4 ; SAVE PCB ADDRESS
05 AE 00000001'9F 51 28 0176 258 MOVC3 R1,@#CTL$T_NODEADDR+1,5(SP) ; COPY REMOTE NODE ADDRESS
      54 8ED0 017F 259 POPL R4 ; RESTORE PCB ADDRESS
      51 5E D0 0182 260 MOVL SP,R1 ; REMOTE NODE ADDRESS ADDRESS
      7E 7C 0185 261 60$: CLRQ -(SP) ; NODE ADDRESS, NAME, REM. ID, IMAGE
      0185 262
      0187 263 ; RESOURCE DATA
      0187 264
      0187 265
55 00000000'9F DD 0187 266 PUSHL @#CTL$GL_VOLUMES ; VOLUMES MOUNTED
      00000000'9F DD 018D 267 MOVL @#CTL$GL_PHD,R5 ; PHD ADDRESS
      7E 54 A5 7D 0194 268 MOVQ PHD$D_DICNT(R5),-(SP) ; DIRECT AND BUFFERED IO COUNTS
7E 00000000'9F 7D 0198 269 MOVQ @#CTL$GL_WSPEAK,-(SP) ; WORKING SET AND PAGE FILE PEAK
      0108 C5 DD 019F 270 PUSHL PHD$D_PGFLTIO(R5) ; PAGE FAULT I/O COUNT
      4C A5 DD 01A3 271 PUSHL PHD$D_PAGEFLTS(R5) ; PAGE FAULT COUNT
      38 A5 DD 01A6 272 PUSHL PHD$D_CPUTIM(R5) ; CPU TIME
      00F4 C5 DD 01A9 273 PUSHL PHD$D_IMGCNT(R5) ; IMAGE EXECUTION COUNT
7E 00000000'9F DD 01AD 274 PUSHL @#CTL$GL_FINALSTS ; FINAL STATUS
      00000000'9F 7D 01B3 275 MOVQ @#CTL$GQ_LOGIN,-(SP) ; LOGIN TIME
      01BA 276
      01BA 277 .DISABLE LSB
```



```
01BA 279
01BA 280
01BA 281 IDENT:
7E 00000000'EF 7D 01BA 282
7E 44 A4 7D 01C1 283
68 A4 DD 01C5 284
24 A4 DD 01C8 285
64 A4 DD 01CB 286
7E B4 01CE 287
7E 94 01D0 288
7E 1F 2F A4 83 01D2 289
7E 00000000'9F 7D 01D7 290
7E 00000004'9F 7D 01DE 291
00000000'9F DD 01E5 292
00BC C4 DD 01EB 293
7E 0084 C4 7D 01EF 294
56 DD 01F4 295
51 D5 01F6 296
07 13 01F8 297
51 5E C2 01FA 298
74 AE 51 F7 01FD 299
10$:
58 D5 0201 300
07 13 0203 301
58 5E C2 0205 302
76 AE 58 F7 0208 303
20$:
59 D5 020C 304
07 13 020E 305
59 5E C2 0210 306
78 AE 59 F7 0213 307
30$:
5A D5 0217 308
07 13 0219 309
5A 5E C2 021B 310
7A AE 5A F7 021E 311
40$:
55 00000000'EF DE 0222 312
53 57 5E C3 0229 313
54 5E D0 022D 314
00000000'GF 16 0230 315
5E 57 D0 0236 316
EXIT:
OFFE 8F BA 0239 317
55 D5 023D 318
0A 12 023F 319
00000000'9F 7C 0241 320
0247 321
05 024A 322
10$:
50 55 D0 024B 323
00000000'EF 17 024E 324
0254 325
0254 326
0254 327
0254 328
0254 329
0254 330
0254 331
0254 332
.END

.SBTTL COMMON IDENTIFICATION DATA

MOVQ EXESGQ_SYSTIME,-(SP) ; CURRENT TIME
MOVQ PCB$T_TERMINAL(R4),-(SP) ; INTERACTIVE PROCESS TERMINAL NAME
PUSHL PCB$L_EOWNER(R4) ; EXTENDED PID OF SUBPROCESS OWNER (0 => NON
PUSHL PCB$L_STS(R4) ; PROCESS STATUS
PUSHL PCB$L_EPID(R4) ; EXTENDED PROCESS ID
CLRW -(SP) ; 3 SPARE BYTES
CLRB -(SP)
SUBB3 PCB$B_PRIOR(R4),#31,-(SP) ; PROCESS BASE PRIORITY
MOVQ @#CTL$T_ACCOUNT,-(SP) ; ACCOUNT NAME
MOVQ @#CTL$T_USERNAME+4,-(SP) ; USERNAME
PUSHL @#CTL$T_USERNAME
PUSHL PCB$L_UIC(R4) ; PROCESS UIC
MOVQ PCB$Q_PRIV(R4),-(SP) ; PRIVILEGE MASK
PUSHL R6 ; MESSAGE TYPE AND RESPONSE MAILBOX (0)
TSTL R1 ; NODE ADDRESS PRESENT ?
BEQL 10$ ; IF EQL, NO
SUBL2 SP,R1 ; NODE NAME OFFSET
CVTLW R1,ACM$W_NODEADDR(SP) ; NODE NAME OFFSET

TSTL R8 ; NODE NAME PRESENT ?
BEQL 20$ ; IF EQL, NO
SUBL2 SP,R8 ; NODE NAME OFFSET
CVTLW R8,ACM$W_NODENAME(SP) ; NODE NAME OFFSET

TSTL R9 ; REMOTE ID PRESENT ?
BEQL 30$ ; IF EQL, NO
SUBL2 SP,R9 ; REMOTE ID OFFSET
CVTLW R9,ACM$W_REMOTEID(SP) ; REMOTE ID OFFSET

TSTL R10 ; IMAGE NAME PRESENT ?
BEQL 40$ ; IF EQL, NO
SUBL2 SP,R10 ; IMAGE NAME OFFSET
CVTLW R10,ACM$W_IMAGENAME(SP) ; IMAGE NAME OFFSET

MOVAL SYS$GL_JOBCTLMB,R5 ; ACCOUNTING MANAGER MAILBOX UCB ADDR
SUBL3 SP,R7,R3 ; MESSAGE SIZE
MOVL SP,R4 ; MESSAGE ADDRESS
JSB G^EXESSENDMSG ; SEND MESSAGE TO JOB CONTROLLER
MOVL R7,SP ; CLEAN UP STACK

POPR #*M<R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>; RESTORE R1-R11
TSTL R5 ; SPECIAL KERNEL AST ROUTINE ?
BNEQ 10$ ; IF NEQ, YES
CLRQ @#CTL$GQ_ISTART ; MARK IMAGE ACCOUNTING INACTIVE
SETIPL #0 ; ENABLE AST DELIVERY
RSB

MOVL R5,R0 ; ADDRESS OF ACB TO DEALLOCATE
JMP EXES$DEANONPAGED ; DEALLOCATE ACB ON THE WAY OUT
```


ACCOUNT
Symbol table

-- ACCOUNTING MESSAGE SUBROUTINES B 14

15-SEP-1984 23:49:17 VAX/VMS Macro V04-00
5-SEP-1984 03:39:51 [SYS.SRC]ACCOUNT.MAR;1

Page 9
(2)

ACMSB_PROCPRI	= 00000024		
ACMSK_PROCLN	= 0000007C		
ACMSL_BIOCNT	= 0000006C		
ACMSL_CPUTIME	= 00000054		
ACMSL_DIOCNT	= 00000068		
ACMSL_FINALSTS	= 0000004C		
ACMSL_IMGCNT	= 00000050		
ACMSL_OWNER	= 00000030		
ACMSL_PAGEFLTS	= 00000058		
ACMSL_PGFLPEAK	= 00000064		
ACMSL_PGFLTIO	= 0000005C		
ACMSL_PID	= 00000028		
ACMSL_STS	= 0000002C		
ACMSL_UIC	= 0000000C		
ACMSL_VOLUMES	= 00000070		
ACMSL_WSPEAK	= 00000060		
ACMSQ_LOGIN	= 00000044		
ACMSQ_PRVMSK	= 00000004		
ACMSQ_SYSTIME	= 0000003C		
ACMST_ACCOUNT	= 0000001C		
ACMST_TERMINAL	= 00000034		
ACMST_USERNAME	= 00000010		
ACMSV_IMAGE	= 00000001		
ACMSW_IMAGENAME	= 0000007A		
ACMSW_MAILBOX	= 00000002		
ACMSW_NODEADDR	= 00000074		
ACMSW_NODENAME	= 00000076		
ACMSW_REMOTEID	= 00000078		
ACMSW_TYPE	= 00000000		
CTL\$GL_FINALSTS	*****	X	01
CTL\$GL_IBIOCNT	*****	X	01
CTL\$GL_ICPUTIM	*****	X	01
CTL\$GL_IDIOCNT	*****	X	01
CTL\$GL_IFAULTIO	*****	X	01
CTL\$GL_IFAULTS	*****	X	01
CTL\$GL_IMGHDRBF	*****	X	01
CTL\$GL_IVOLUMES	*****	X	01
CTL\$GL_IWSPEAK	*****	X	01
CTL\$GL_PHD	*****	X	01
CTL\$GL_VOLUMES	*****	X	01
CTL\$GL_WSPEAK	*****	X	01
CTL\$GQ_ISTART	*****	X	01
CTL\$GQ_LOGIN	*****	X	01
CTL\$T_ACCOUNT	*****	X	01
CTL\$T_NODEADDR	*****	X	01
CTL\$T_NODENAME	*****	X	01
CTL\$T_REMOTEID	*****	X	01
CTL\$T_USERNAME	*****	X	01
EXESCHKIMAGENAME	*****	X	01
EXESDEANONPAGED	*****	X	01
EXESGL_ACMFLAGS	*****	X	01
EXESGQ_SYSTIME	*****	X	01
EXESIMGDELMSG	00000009	RG	01
EXESIMGPURMSG	00000000	RG	01
EXESPRCDELMSG	00000111	RG	01
EXESPRCPURMSG	00000108	RG	01
EXESSENDMSG	*****	X	01

EXIT	00000239	R	01
IDENT	000001BA	R	01
IPL\$ASTDEL	= 00000002		
MSG\$DELIMAG	= 0000000C		
MSG\$DELPROC	= 00000003		
MSG\$PURIMAG	= 0000000D		
MSG\$PURPROC	= 0000000B		
PCBSB_PRIIB	= 0000002F		
PCBSL_EOWNER	= 00000068		
PCBSL_EPID	= 00000064		
PCBSL_STS	= 00000024		
PCBSL_UIC	= 000000BC		
PCBSQ_PRIV	= 00000084		
PCBST_TERMINAL	= 00000044		
PCBSV_NOACNT	= 0000000F		
PHDSL_BIOCNT	= 00000058		
PHDSL_CPUTIM	= 00000038		
PHDSL_DIOCNT	= 00000054		
PHDSL_IMGCNT	= 000000F4		
PHDSL_PAGEFLTS	= 0000004C		
PHDSL_PGFLTIO	= 00000108		
PR\$ IPL	*****	X	01
SYS\$GL_JOBCTLMB	*****	X	01

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes															
ABS	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE					
YF\$LOWUSE	00000254 (596.)	01 (1.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE					
\$ABSS	00000000 (0.)	02 (2.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE					

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	34	00:00:00.07	00:00:00.50
Command processing	164	00:00:00.64	00:00:04.70
Pass 1	211	00:00:04.94	00:00:20.84
Symbol table sort	0	00:00:00.58	00:00:01.93
Pass 2	77	00:00:01.19	00:00:04.47
Symbol table output	10	00:00:00.09	00:00:00.35
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	500	00:00:07.54	00:00:32.81

The working set limit was 1350 pages.
26742 bytes (53 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 423 non-local and 18 local symbols.
332 source lines were read in Pass 1, producing 16 object records in Pass 2.
14 pages of virtual memory were used to define 13 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	5
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	5
TOTALS (all libraries)	10

479 GETS were required to define 10 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:ACCOUNT/OBJ=OBJ\$:ACCOUNT MSRC\$:ACCOUNT/UPDATE=(ENH\$:ACCOUNT)+EXECML\$/LIB

0372

AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY